UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/813,433	03/31/2004	Simon Knowles	321546US	3801

27964 7590 08/24/2010
HITT GAINES P.C.
P.O. BOX 832570
RICHARDSON, TX 75083

EXAMINER	
HUISMAN, DAVID J	
ART UNIT	**PAPER NUMBER**
2183	

NOTIFICATION DATE	DELIVERY MODE
08/24/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket@hittgaines.com

Office Action Summary	Application No.	Applicant(s)	
	10/813,433	KNOWLES, SIMON	
	Examiner	Art Unit	
	DAVID J. HUISMAN	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _10 June 2010_.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-22 and 30_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-22 and 30_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _13 March 2004_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _7/29/10_.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-22 and 30 have been examined.

Information Disclosure Statement

2. The document cited in the IDS submitted on July 29, 2010, has been struck through and not considered because the citation does not include the author, which is required by 37 CFR 1.98(b)(5).

Claim Objections

3. Claim 1 is objected to because of the following informalities:

- In line 7, it appears that the commas around "comprising" are unnecessary.
- In line 1 of the 2nd to last paragraph, replace "a said instruction" with --one of said control instructions--.
- Either replace both occurrences of "a" with --said-- on page 2, last line, and page 3, first line, or in the 2nd to last line, replace "said" with --a--.

4. Claim 22 is objected to because of the following informalities:

- In line 5, it appears that the comma after "comprising" should be removed.

5. Claim 30 is objected to because of the following informalities:

- In line 7, it appears that the commas around "comprising" are unnecessary.
- Some form of punctuation is needed at the end of the 3rd to last paragraph.
- In line 1 of the 2nd to last paragraph, replace "a said instruction" with --one of said control instructions--.

- In the last two lines, there is a lack of antecedent basis for "said configurable data processing instruction". Is applicant referring to the configurable data processing instruction of lines 15-16, or of line 20? Either replace both occurrences of "a" with --said-- on page 3, 1st-2nd lines, or in the 2nd to last line, replace "said" with --a--.

Appropriate correction is required.

Double Patenting

6. Applicant is advised that should claim 30 be found allowable, claim 16 will be objected to under 37 CFR 1.75 as being a substantial duplicate thereof (note that claim 16 includes all limitations of claim 1). When two claims in an application are duplicates or else are so close in content that they both cover the same thing, despite a slight difference in wording, it is proper after allowing one claim to object to the other as being a substantial duplicate of the allowed claim. See MPEP § 706.03(k).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1-22 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Trimberger, U.S. Patent No. 5,737,631, in view of Haynes et al., "Configurable Multiplier

Blocks for use within an FPGA", 1998 (herein referred to as Haynes), Lodi et al., "A Flexible LUT-Based Carry Chain for FPGAs", 2003 (herein referred to as Lodi), and Madurawe, U.S. Patent No. 7,176,713.

9. Referring to claim 1, Trimberger has taught a hardware computer processor having control and data processing capabilities comprising:

a) a hardware decode unit for decoding instructions and operable to separate control instructions from data processing instructions (Trimberger: Figure 2, item 112). Control instructions for execution unit 100 and data processing instructions for FPGA 120 are inherently separated because the purpose of the decoder is to identify the type of instruction and control the appropriate circuitry to carry out the operation indicated by the instruction. Therefore, if a control instruction exists, appropriate control signals would be sent to unit 100. If a data processing instruction exists, appropriate control signals would be sent to FPGA 120.

b) a dedicated hardware control processing facility comprising a control execution path dedicated to processing said control instructions having its own control register file (Fig.2, component 103 or 140, for instance) and a hardware execution unit (Fig.2, component 100). Note that <u>at least</u> these components may be combined and called a dedicated control processing facility.

c) a dedicated hardware data processing facility (Fig.2, at least component 120) dedicated to processing said data processing instructions, separate from said dedicated control processing facility, having its own data register file (Fig.2, <u>at least</u> component 130) separate from said control register file, the data processing facility comprising a controller (controllers inherently exist in processors).

d) Trimberger has not taught that the data processing facility comprises a first data execution path including fixed operators and a second data execution path including at least configurable operators, both of said first and second data execution paths separate from said control execution path and each other, said configurable operators pre-configured into a plurality of hardwired operator classes. However, Haynes, Lodi, and Madurawe have taught that FPGAs may be designed to include dedicated circuitry in addition to reconfigurable circuitry. Specifically, Haynes has taught dedicated multiplier blocks, which may be programmably interconnected to create larger multipliers. Similarly, Lodi has taught dedicated carry chains, which may be connected to form, among other things, absolute value and logic function circuitry. See page 133 and sections 3.3 and 4. Madurawe has taught dedicated adders and I/O circuitry in an FPGA for common operations. See column 6, lines 2-4. Essentially, instead of a fully reprogrammable FPGA, as taught by Trimberger, Haynes, Lodi, and Madurawe have taught programmable devices, such as FPGAs, that include fixed and configurable circuitry. Such a configuration, which is common in a Xilinx Virtex FPGA, for instance, allows programmers to achieve balance between programmability and speed for different applications. One of ordinary skill in the art would have recognized that the hardwired operators of Haynes, Lodi, and Madurawe could be implemented in the FPGA of Trimberger. Such a modification would allow Trimberger to achieve increased speed in cases where dedicated circuitry is faster than programmable circuitry. As a result, in order to increase speed for common operations, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Trimberger's gate array such that the data processing facility (Trimberger, Fig.2, at least component 120) comprises a first data execution path including fixed operators (for instance, a fixed adder and I/O path, as taught

by Madurawe) and a second data execution path including at least configurable operators (for instance, a configurable multiplier and carry-chain path, as taught by Haynes and Lodi), both of said first and second data execution paths separate from said control execution path and each other (clearly the FPGA paths are separate from each other and from the control path), said configurable operators pre-configured into a plurality of hardwired operator classes (in the proposed combination, the FPGA of Trimberger, as modified, would include configurable operators pre-configured into hardwired multiplier and carry chain classes).

d) Trimberger has further taught that said decode unit is operable to supply a said control instruction to a functional unit in said dedicated control processing facility and operable to detect whether one of said data processing instructions defines a fixed data processing instruction or a configurable data processing instruction, said decode unit causing the computer processor to supply said data processing instruction to said first data execution path for processing when a fixed data processing instruction is detected and to said second data execution path for processing when a configurable data processing instruction is detected. See Fig.2, component 112 and column 7, lines 45-50. The examiner asserts that decoders inherently operation in the claimed fashion. They distinguish all types of instructions from each other, and produce the appropriate control signals 113 to control the appropriate circuitry to perform the desired operation.

e) Trimberger has not taught that the dedicated control processing facility also includes functional units comprising a branch unit (despite hinting at existence of a branch unit with disclosure of condition codes (Fig.2)) and a load/store unit. However, the examiner asserts that such units are very well known in the art and are advantageous for providing basic functionality.

Branch units are clearly advantageous because they allow for execution of branch instructions. These are at least useful so that loops may be implemented. Loops allow for repeated execution of a block of code without writing the code repetitively within the program. Hence, code size may be decreased. Load and stores, on the other hand, allow for communication data to/from the register file and memory, thereby expanding the usable memory space. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Trimberger such that the dedicated control processing facility includes a branch unit and load/store unit.

f) Trimberger, as modified, has further taught that said controller is operable to configure the connectivity of said configurable operators in accordance with configuration information provided in an opcode portion of said configurable data processing instruction. See column 3, lines 31-33, and note that Trimberger would still select the connectivity of the operator classes via program instruction opcode.

10. Referring to claim 2, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, wherein the decode unit is capable of decoding a stream of instruction packets from memory, each packet comprising a plurality of instructions (Trimberger: column 7, lines 51-56).

11. Referring to claim 3, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, wherein the decode unit is operable to detect if an instruction packet contains a data processing instruction (Trimberger: column 7, lines 45-50).

12. Referring to claim 4, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, wherein the configurable operators are configurable at the level of multi-bit values (Trimberger: column 9, lines 18-19) (The opcode is a multi-bit value).

13. Referring to claim 5, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 4, wherein the configurable operators are configurable at the level of multi-bit values comprising four or more bits (Trimberger: column 9, lines 18-19) (The opcode is at least 4 bits).

14. Referring to claim 6, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 4, wherein the configurable operators are configurable at the level of words (Trimberger: column 9, lines 24-25) (Immediate values are optional; therefore, the whole word is configurable).

15. Referring to claim 7, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1. Trimberger has not taught that a plurality of the fixed operators of the first data execution path is arranged to perform a plurality of fixed operations in independent lanes according to single instruction multiple data principles. However, Official Notice is taken that SIMD and the related advantages are well known and accepted in the art. Specifically, SIMD allows each execution unit to perform the same instruction on different data in the same cycle, thereby increasing data level parallelism. Higher parallelism will potentially result in higher throughput as more operations can occur at once. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Trimberger such that a plurality of the fixed operators of the first data execution path (Fig.2, component 100)

is arranged to perform a plurality of fixed operations in independent lanes according to single instruction multiple data principles.

16. Referring to claim 8, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1. Trimberger in view of Haynes and Lodi has not taught that a plurality of the configurable operators of the second data execution path is arranged to perform multiple operations in different lanes according to single instruction multiple data principles. However, Official Notice is taken that SIMD and the related advantages are well known and accepted in the art. Specifically, SIMD allows each execution unit to perform the same instruction on different data in the same cycle, thereby increasing data level parallelism. Higher parallelism will potentially result in higher throughput as more operations can occur at once. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Trimberger in view of Haynes and Lodi such that a plurality of the configurable operators of the second data execution path (Trimberger, Fig.2, component 120) is arranged to perform multiple operations in different lanes according to single instruction multiple data principles.

17. Referring to claim 9, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, wherein configurable operators of the second execution path are arranged to receive configuration information which determines the nature of the operations performed (Trimberger: column 8, lines 5-17).

18. Referring to claim 10, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 9, wherein configurable operators of the second execution path are arranged to receive configuration information which determines the nature of the operations

performed from a field of an instruction defining a configurable data processing operation (Trimberger: column 7, lines 62-67; column 8, lines 1-17).

19. Referring to claim 11, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, wherein configurable operators of the second execution path are arranged to receive configuration information comprising information controlling connectivity of the configurable operators (Trimberger: column 8, lines 35-37).

20. Referring to claim 12, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 9, comprising a control map associated with configurable operators of the second data execution path, said control map being operable to receive at least one configuration bit from a configurable data processing instruction and to provide configuration information to the configurable operators responsive thereto (Trimberger: column 7, lines 62-67; column 8, lines 1-17).

21. Referring to claim 13, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 12, wherein said configuration information controls interconnectivity between two or more of said configurable operators (Trimberger: column 8, lines 35-37).

22. Referring to claim 14, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, wherein configurable operators of the second execution path are arranged to receive either configuration information determining the nature of an operation to be performed or configuration information controlling interconnectivity from a source other than a configurable data processing instruction (Trimberger: column 8, lines 5-17; Figure 2; items 101, 102 and 123).

23. Referring to claim 15, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, wherein at least one configurable operator of the second data execution path is capable of executing data processing instructions with an execution depth greater than two computations before returning results to a results store (Trimberger: column 3, lines 10-27) (It is inherent that these complex functions will take at least two cycles).

24. Referring to claim 16, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, comprising a switch mechanism for receiving data processing operands from a configurable data processing instruction and switching them as appropriate for supply to one or more of said configurable operators (Trimberger: column 7, lines 45-50).

25. Referring to claim 17, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, comprising a switch mechanism for receiving results from one or more of said configurable operators and switching the results as appropriate for supply to one or more of a result store and feed back loop (Trimberger: column 8, lines 51-59).

26. Referring to claim 18, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, comprising a plurality of control maps for mapping configuration bits received from configurable data processing instructions to configuration information for supply to configurable operators of the second data execution path (Trimberger: column 3, lines 66-67; column 4, lines 1-10).

27. Referring to claim 19, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, comprising a switch mechanism for receiving configuration information from a control map and switching it as appropriate for supply to configurable operators of the second data execution path (Trimberger: column 8, lines 5-17).

28. Referring to claim 20, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, comprising configurable operators selected from one or more of: multiply accumulate operators; arithmetic operators; state operators; and cross-lane permuters (Trimberger: column 3, lines 10-27).

29. Referring to claim 21, Trimberger in view of Haynes and Lodi has taught a computer processor according to claim 1, comprising operators and an instruction set capable of performing one or more operations selected from: Fast Fourier Transforms; Inverse Fast Fourier Transforms; Viterbi encoding/decoding; Turbo encoding/decoding; and Finite Impulse Response calculations; and any other Correlations or Convolutions (Trimberger: column 3, lines 10-27) (polynomial evaluation is used in FFT and IFFT).

30. Referring to claim 22, Trimberger has taught a method of operating a computer processor having control and data processing capabilities, said computer processor comprising:

a) a decode unit for decoding instructions (Fig.2, component 112).

b) a dedicated control processing facility comprising a control execution path dedicated to processing control instructions having its own control register file (Fig.2, component 103 or 140, for instance) and an execution unit (Fig.2, component 100). Note that <u>at least</u> these components may be combined and called a dedicated control processing facility.

c) a dedicated data processing facility (Fig.2, at least component 120) dedicated to processing data processing instructions, separate from said dedicated control processing facility, having its own data register file (Fig.2, at least 130) separate from said control register file, the data processing facility comprising a controller (controllers inherently exist in processors).

d) Trimberger has not taught that the data processing facility comprises a first data execution path including fixed operators and a second data execution path including at least configurable operators, both of said first and second data execution paths separate from said control execution path and each other, said configurable operators pre-configured into a plurality of hardwired operator classes. However, Haynes, Lodi, and Madurawe have taught that FPGAs may be designed to include dedicated circuitry in addition to reconfigurable circuitry. Specifically, Haynes has taught dedicated multiplier blocks, which may be programmably interconnected to create larger multipliers. Similarly, Lodi has taught dedicated carry chains, which may be connected to form, among other things, absolute value and logic function circuitry. See page 133 and sections 3.3 and 4. Madurawe has taught dedicated adders and I/O circuitry in an FPGA for common operations. See column 6, lines 2-4. Essentially, instead of a fully reprogrammable FPGA, as taught by Trimberger, Haynes, Lodi, and Madurawe have taught programmable devices, such as FPGAs, that include fixed and configurable circuitry. Such a configuration, which is common in a Xilinx Virtex FPGA, for instance, allows programmers to achieve balance between programmability and speed for different applications. One of ordinary skill in the art would have recognized that the hardwired operators of Haynes, Lodi, and Madurawe could be implemented in the FPGA of Trimberger. Such a modification would allow Trimberger to achieve increased speed in cases where dedicated circuitry is faster than programmable circuitry. As a result, in order to increase speed for common operations, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Trimberger's gate array such that the data processing facility (Trimberger, Fig.2, at least component 120) comprises a first data execution path including fixed operators (for instance, a fixed adder and I/O path, as taught

by Madurawe) and a second data execution path including at least configurable operators (for instance, a configurable multiplier and carry-chain path, as taught by Haynes and Lodi), both of said first and second data execution paths separate from said control execution path and each other (clearly the FPGA paths are separate from each other and from the control path), said configurable operators pre-configured into a plurality of hardwired operator classes (in the proposed combination, the FPGA of Trimberger, as modified, would include configurable operators pre-configured into hardwired multiplier and carry chain classes).

e) Trimberger has further taught that the method comprises separating, with said decode unit, control instructions from data processing instructions. See Fig.2, component 112 and column 7, lines 45-50. The examiner asserts that the decoder inherently operates as claimed as the whole purpose of a decoder is to determine the type of instruction and generate the appropriate control signals 113 to control the appropriate circuitry to perform the desired operation. Clearly, if the instruction is meant for execution unit 100, then an instruction for that unit will be determined and signals for that unit will be supplied by the decoder.

f) Trimberger has further taught supplying, by said decode unit, one of said control instructions to a functional unit in said dedicated control processing facility. See Fig.2. The decoder will send control instructions (any instruction that may be executed by the dedicated control processing facility) to the dedicated control processing facility.

g) Trimberger has further taught that the method comprises decoding a plurality of instructions to detect whether at least one of said data processing instructions of said plurality of instructions defines a fixed data processing instruction or a configurable data processing instruction. See Fig.2, component 112. This is again deemed inherent. If an instruction is a fixed path

instruction, then appropriate signals will be sent to the fixed FPGA circuitry. If the instruction is a configurable instruction, then appropriate signals will be sent to the configurable FPGA circuitry.

h) Trimberger has further taught that the method comprises causing the computer processor to supply said at least one data processing instruction to said first data execution path for processing when a fixed data processing instruction is detected and to said second data execution path for processing when a configurable data processing instruction is detected; and outputting results produced by said first data execution path when a fixed data processing instruction is detected and outputting results produced by said data execution path when a configurable processing instruction is detected. This is deemed inherent. Clearly, if the FPGA is to perform a particular add, then the fixed adder will be used to execute the add and produce a result. Similarly, if an absolute value is to be performed, the configurable carry-chain will be used to produce a result.

i) Trimberger has not taught that the dedicated control processing facility also includes functional units comprising a branch unit and a load/store unit. However, the examiner asserts that such units are very well known in the art and are advantageous for providing basic functionality. Branch units are clearly advantageous because they allow for execution of branch instructions. These are at least useful so that loops may be implemented. Loops allow for repeated execution of a block of code without writing the code repetitively within the program. Hence, code size may be decreased. Load and stores, on the other hand, allow for communication data to/from the register file and memory, thereby expanding the usable memory space. As a result, it would have been obvious to one of ordinary skill in the art at the time of the

invention to modify Trimberger such that the dedicated control processing facility includes a branch unit and load/store unit.

j) Trimberger, as modified, has further taught configuring the connectivity of said configurable operators in accordance with configuration information provided in an opcode portion of said configurable data processing instruction. See column 3, lines 31-33, and note that Trimberger would still select the connectivity of the operator classes via program instruction opcode.

31. Referring to claim 30, Trimberger has taught a hardware computer processor having control and data processing capabilities comprising:

a) a hardware decode unit for decoding instructions and operable to separate control instructions from data processing instructions (Trimberger: Figure 2, item 112). Control instructions for execution unit 100 and data processing instructions for FPGA 120 are inherently separated because the purpose of the decoder is to identify the type of instruction and control the appropriate circuitry to carry out the operation indicated by the instruction. Therefore, if a control instruction exists, appropriate control signals would be sent to unit 100. If a data processing instruction exists, appropriate control signals would be sent to FPGA 120.

b) a dedicated hardware control processing facility comprising a control execution path dedicated to processing said control instructions having its own control register file (Fig.2, component 103 or 140, for instance) and a hardware execution unit (Fig.2, component 100). Note that <u>at least</u> these components may be combined and called a dedicated control processing facility.

c) a dedicated hardware data processing facility (Fig.2, at least component 120) dedicated to processing said data processing instructions, separate from said dedicated control processing facility, having its own data register file (Fig.2, <u>at least</u> component 130) separate from said

control register file, the data processing facility comprising a controller (controllers inherently exist in processors).

d) Trimberger has not taught that the data processing facility comprises a first data execution path including fixed operators and a second data execution path including at least configurable operators, both of said first and second data execution paths separate from said control execution path and each other, said configurable operators pre-configured into a plurality of hardwired operator classes. However, Haynes, Lodi, and Madurawe have taught that FPGAs may be designed to include dedicated circuitry in addition to reconfigurable circuitry. Specifically, Haynes has taught dedicated multiplier blocks, which may be programmably interconnected to create larger multipliers. Similarly, Lodi has taught dedicated carry chains, which may be connected to form, among other things, absolute value and logic function circuitry. See page 133 and sections 3.3 and 4. Madurawe has taught dedicated adders and I/O circuitry in an FPGA for common operations. See column 6, lines 2-4. Essentially, instead of a fully reprogrammable FPGA, as taught by Trimberger, Haynes, Lodi, and Madurawe have taught programmable devices, such as FPGAs, that include fixed and configurable circuitry. Such a configuration, which is common in a Xilinx Virtex FPGA, for instance, allows programmers to achieve balance between programmability and speed for different applications. One of ordinary skill in the art would have recognized that the hardwired operators of Haynes, Lodi, and Madurawe could be implemented in the FPGA of Trimberger. Such a modification would allow Trimberger to achieve increased speed in cases where dedicated circuitry is faster than programmable circuitry. As a result, in order to increase speed for common operations, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Trimberger's gate array such

that the data processing facility (Trimberger, Fig.2, at least component 120) comprises a first data execution path including fixed operators (for instance, a fixed adder and I/O path, as taught by Madurawe) and a second data execution path including at least configurable operators (for instance, a configurable multiplier and carry-chain path, as taught by Haynes and Lodi), both of said first and second data execution paths separate from said control execution path and each other (clearly the FPGA paths are separate from each other and from the control path), said configurable operators pre-configured into a plurality of hardwired operator classes (in the proposed combination, the FPGA of Trimberger, as modified, would include configurable operators pre-configured into hardwired multiplier and carry chain classes).

d) Trimberger has further taught a switch mechanism for receiving data processing operands from a configurable data processing instruction and switching them as appropriate for supply to one or more of said configurable operators. See, Trimberger, column 7, lines 45-50. When an instruction calls for an operation using the configurable operators, then operands will be supplied to them. For instance, as described above, multipliers may be one group of configurable operators. To perform multiplication, multiple input values (operands) are needed. These operands are therefore switched to the operators for execution.

e) Trimberger has further taught that said decode unit is operable to supply a said control instruction to a functional unit in said dedicated control processing facility and operable to detect whether one of said data processing instructions defines a fixed data processing instruction or a configurable data processing instruction, said decode unit causing the computer processor to supply said data processing instruction to said first data execution path for processing when a fixed data processing instruction is detected and to said second data execution path for

processing when a configurable data processing instruction is detected. See Fig.2, component 112 and column 7, lines 45-50. The examiner asserts that decoders inherently operation in the claimed fashion. They distinguish all types of instructions from each other, and produce the appropriate control signals 113 to control the appropriate circuitry to perform the desired operation.

f) Trimberger has not taught that the dedicated control processing facility also includes functional units comprising a branch unit (despite hinting at existence of a branch unit with disclosure of condition codes (Fig.2)) and a load/store unit. However, the examiner asserts that such units are very well known in the art and are advantageous for providing basic functionality. Branch units are clearly advantageous because they allow for execution of branch instructions. These are at least useful so that loops may be implemented. Loops allow for repeated execution of a block of code without writing the code repetitively within the program. Hence, code size may be decreased. Load and stores, on the other hand, allow for communication data to/from the register file and memory, thereby expanding the usable memory space. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Trimberger such that the dedicated control processing facility includes a branch unit and load/store unit.

g) Trimberger, as modified, has further taught that said controller is operable to configure the connectivity of said configurable operators in accordance with configuration information provided in an opcode portion of said configurable data processing instruction. See column 3, lines 31-33, and note that Trimberger would still select the connectivity of the operator classes via program instruction opcode.

Response to Arguments

32. Applicant's arguments filed on June 10, 2010, have been fully considered but they are not persuasive.

33. Applicant argues the novelty/rejection of claims 1 and 22 on pages 11-15 of the remarks, in substance that:

> (1) "As noted above, independent Claims 1 and 22 have been amended to more clearly point out that the control execution path of the dedicated control processing facility is dedicated to processing control instructions and also that the dedicated control processing facility comprises functional units. Trimberger only has a single execution unit 100.
>
> Furthermore, execution unit 100 and its registers 140 are not dedicated to the processing of control instructions. As stated in line 65 of column 5 of Trimberger, the execution unit provides a general purpose microprocessor. That is, this execution unit handles all instructions in the program, apart from the instructions which will be handled by RISA 21.
>
> Additionally, Claim 1 has been amended, as noted above, to more clearly point out that the decode unit is operable to separate control instructions from data processing instructions. At item 8d on page 5 of the Office Action, the Examiner addresses this. The decoder 112 in Trimberger does not separate different classes of instructions. Instead, it supplies the opcode from the instructions to the instruction input I on the defined execution unit 100 and to the instruction port I on the RISA FPGA 120. It is clear that the decoder 112 does not separate the instructions into different classes. In fact, in Trimberger, there is an instruction format in which a defined instruction opcode field and a programmed instruction opcode field are present in the same instruction where these opcodes are simultaneously applied to the execution unit 100 and the RISA FPGA 120. (See, e.g., Fig. 5 and lines 41-57 of column 9 of Trimberger.) There is clearly no separation of the instructions for supply to separate dedicated processing paths as recited in presently amended independent Claims 1 and 22."
>
> (2) "...the Examiner recognizes that Trimberger does not disclose that the data processing facility comprises a first data execution path and a second data execution path but cites that there is some combination of Haynes, Lodi, and Madurawe which would result in such a configuration. There is no basis on which a person of ordinary skill in the art at the time of the invention, without the benefit of the teachings of the present invention, could derive such a combination of these documents without improper hindsight."
>
> (3) "...In Trimberger, however, these condition codes are utilized by an instruction control state machine 107 which manages the instruction sequencing decisions. (See, e.g., lines 35-38 of column 7 of Trimberger.) It appears these instruction sequencing decisions are handled for the whole system and, therefore, there is no motivation in Trimberger to include a separate branch unit which is dedicated just to the execution unit 100 (equated as the claimed dedicated control processing facility)."
>
> (4) "...the Examiner acknowledges that Trimberger does not teach that the dedicated control processing facility includes a load/store unit. Loading and storing is accomplished in Trimberger using external ports connected to external memory as shown on the right and left hand sides of Fig. 2. There would be no need to include a separate load/store unit dedicated only to the

> execution unit 100 because the ports in Trimberger perform the necessary task of loading and storing to and from external memory for all of the units in the system."
>
> (5) "...This part of Trimberger suggests that the RISA can be reprogrammed by a program. That is, instructions can be provided which reprogram the RISA. However, Trimberger does not disclose that the data processing instructions themselves include the configuration information in their opcode portion. The opcode portions of the instructions of Trimberger are shown in Figs. 3-5 and emphatically do not contain any configuration information.
>
> Configuration of the RISA is done as a separate matter, as discussed in lines 39-44 of column 8 of Trimberger. While the RISA is being configured, it cannot be used to execute instructions. Thus, Trimberger cannot and does not disclose data processing instructions which are supplied to the data execution path for processing and which include configuration information in their opcode portion."
>
> (6) "Furthermore, the Examiner suggests that Trimberger teaches the supply of a fixed path instruction to "fixed FPGA circuitry" and a configurable instruction to "configurable FPGA circuitry." There is no disclosure in Trimberger of "fixed FPGA circuitry." The entire FPGA 120 of Trimberger is configurable."

34. These arguments are not found persuasive for the following reasons:

a) Regarding the first argument, as explained in the rejection, it would have been obvious to modify the control processing facility to include multiple functional units (such as a branch unit and load/store unit). Hence, Trimberger, as modified, does not just have a single execution unit in the control processing facility. There would be at least a branch unit, a load/store unit, and a unit for executing general arithmetic/logic instructions, as known in the art. The control processing facility is dedicated to processing control instructions, which under one interpretation, may be considered to be all instructions not executed by FPGA 120. Under a second interpretation, control instructions may be considered to be branch instructions, which would then be executed by the branch unit in the control processing facility. Either way, because the control processing facility executes instructions the data processing facility does not, it is dedicated to executing those instructions. Regarding the decoder, the examiner asserts that separation is an inherent function performed by the decoder. The purpose of a decoder is to identify a type of instruction and send control signals to corresponding circuitry to carry out the

operation set forth by the instruction. Hence, it must distinguish between (i.e., separate) control and data processing instructions, and send control signals to either the control of data processing facility based on the type of instruction. A similar response applies to the data processing facility arguments on pages 12-13 of the remarks.

b) Regarding the second argument, the examiner disagrees that there is no basis for combination. As shown in the secondary references, fixed and configurable logic may co-exist on an FPGA. This is done for known reasons, as described in the rejection above. Hence, the examiner asserts that the combination is valid.

c) Regarding the third argument, the examiner modified the control processing facility of Trimberger to include a branch unit because the control processing facility is generally compared to a general purpose microprocessor (column 5, line 65). Branch processing is very well known in microprocessors. Also, the FPGA generally exists to execute more complicated, time-consuming instructions, which don't match well with pipelining (column 6, lines 52-58). Branches do not fall into this category of instruction.

d) Regarding the fourth argument, as is known, load/store instructions are executed. Hence, there must be some functional unit to execute them. The examiner asserts that it is obvious for the load/store unit to be in the control processing facility, especially given the fact that the control processing facility is compared to a general purpose microprocessor. Most, if not all, microprocessors execute load/store instructions. While memory ports are utilized to carry data between memory and the processor in response to a load/store, this does not mean that an execution unit does not exist for executing the load/store instructions.

e) Regarding the fifth argument, it is clear from column 3, lines 31-33, that Trimberger is concerned with dynamic reconfiguration, in at least one embodiment. For a program to reconfigure the processor, at least one instruction must be executed (as all actions in a processor occur in response to instructions). Therefore, for reconfiguration to occur, an instruction must be identified as a configuration instruction. In order to identify a configuration instruction, an opcode portion must be used to specify that the instruction is a configuration-type instruction. It should also be noted that while the FPGA 120 may be halted while being reconfigured, this does not mean that it cannot execute the configuration instruction to initiate reconfiguration.

f) Regarding the sixth argument, applicant should note that Trimberger's FPGA was modified to include both fixed and configurable circuitry, as taught by the secondary references. This modification is obvious for reasons set forth in the rejection above.

Conclusion

35. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DAVID J. HUISMAN whose telephone number is (571)272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/David J. Huisman/
Primary Examiner, Art Unit 2183